Dottorato di Ricerca in Informatica - Ciclo XXII
Dipartimento di Informatica, Sistemistica e Comunicazione
Facoltà di Scienze Matematiche, Fisiche e Naturali
Università degli Studi di Milano–Bicocca

Pure Parsimony Xor-Haplotyping

Elaborato finale del corso di dottorato
"Teoria ed Applicazione del Calcolo Evoluzionistico"

Docente: Leonardo Vanneschi

6 marzo 2008

Yuri Pirola

Outline

Scenario biologico

- Scenari:
  - Malattia X rara e incurabile
  - Farmaco Y per la patologia Z inefficace *solo* su alcune persone
- Domanda: perché?
- Ipotesi: possibile influenza di fattori genetici $\longrightarrow$ da verificare!
- Verifica: **Studi di Associazione**

  Input: Patrimonio genetico degli individui di una popolazione e una (o più) caratteristica osservabile ad essi associata.
  Output: "Correlazioni" fra variazioni genetiche fra gli individui e lo stato della caratteristica osservata.

Patrimonio Genetico

- Ottenere il patrimonio genetico degli individui di una popolazione è dispendioso
- Scorciatoia: ottenere i genotipi di tutti gli individui e risolverli "computazionalmente" in aplotipi $\longrightarrow$ *haplotyping*

Esempio:

Carattere:	occhi	capelli	carnagione
Genotipo:	$\{c, s\}$	$\{c, s\}$	$\{c, c\}$
Aplotipo materno:	c	s	c
Aplotipo paterno:	s	c	c

Xor-Genotypes Shamir et al., IEEE/ACM Trans. Comp. Biol. Bioinf. 2008

Carattere:	occhi	capelli	carnagione
Genotipo:	$\{c, s\}$	$\{c, s\}$	$\{c, c\}$
Apl. materno:	c	s	c
Apl. paterno:	s	c	c

Osservazioni:

- L'eterozigosi genera ambiguità
- La risoluzione deve essere compiuta coerentemente a un *modello genetico di riferimento*
- **Xor-Genotypes:** un vettore binario in cui l'elemento al posto i è uguale a 1 se il sito i è *eterozigote*, a 0 se il sito i è *omozigote* (e.g. $x = (1\ 1\ 0)$).

Perché Xor-Genotipi?

Sito:	s_1	s_2	s_3
Xor-Genotipo:	1	1	0
Apl. materno:	0	1	0
Apl. paterno:	1	0	0

- Codifica delle varianti alleliche negli aplotipi su alfabeto binario.
- Due aplotipi h_1 e h_2 "risolvono" uno xor-genotipo x se e solo se $h_1 \oplus h_2 = x$.
- Nota: L'insieme dei vettori binari di lunghezza m con l'operazione $\oplus$ di xor è un *gruppo abeliano*.

Pure Parsimony Xor-Haplotyping

Pure Parsimony Xor-Haplotyping (PPXH)

Input: Insieme X di n xor-genotipi di lunghezza m.
Output: Il (un) minimo insieme H di aplotipi che spiega tutti i genotipi in X.

- L'insieme X è rappresentato come matrice binaria $n \times m$.
- L'insieme H è rappresentato come matrice binaria $2n \times m$.
- $H \in \mathcal{S}(\text{PPXH}, X)$ sse $h_{2i} \oplus h_{2i+1} = x_i$ per $0 \leq i < n$.
- Problema di *minimizzazione* con $c(H) :=$ n. di righe differenti in H.
- Complessità computazionale: forse APX-hard, FPT e alcune restrizioni in P.

PPXH e Algoritmi Genetici

- La complessità computazionale giustifica l'uso degli AG.
- Come valutare la bontà dei risultati se non si conosce l'ottimo?
- È necessario un algoritmo di **costruzione di istanze artificiali di costo ottimo noto**.

Data X una matrice $n \times m$ di xor-genotipi.

$$\forall H \in \mathcal{S}(\text{PPXH}, X), c(H) \geq \frac{\sqrt{8n+1}+1}{2} \sim \sqrt{n}$$

Algoritmo di costruzione: genero h aplotipi differenti di lunghezza m, ne costruisco tutte le $n = \frac{h \cdot (h-1)}{2}$ coppie e da esse ottengo i genotipi (della matrice X).

Codifica delle soluzioni

- Individuo = matrice binaria H degli aplotipi.
- Codifica = "concatenazione" delle righe di H in un singolo vettore di lunghezza $2mn$.
- Fitness $f(H) := c(H) :=$ n. di righe differenti in H.
- Problema: non tutti gli individui sono soluzioni ammissibili!
- Soluzione: penalty, aggiungo $+2$ alla fitness per ogni genotipo non risolto $\quad f(H) := c(H) + 2 \cdot$ (n. di errori di ricostruzione).
- Proprietà: per ogni individuo H t.c. $H \notin \mathcal{S}(X)$ esiste $r(H) \in \mathcal{S}(X)$ calc. in tempo polinomiale t.c. $c(r(H)) \leq f(H)$.
- $f(H)$ è una *stima pessimistica* di $c(r(H))$.

Algoritmo genetico: parametri

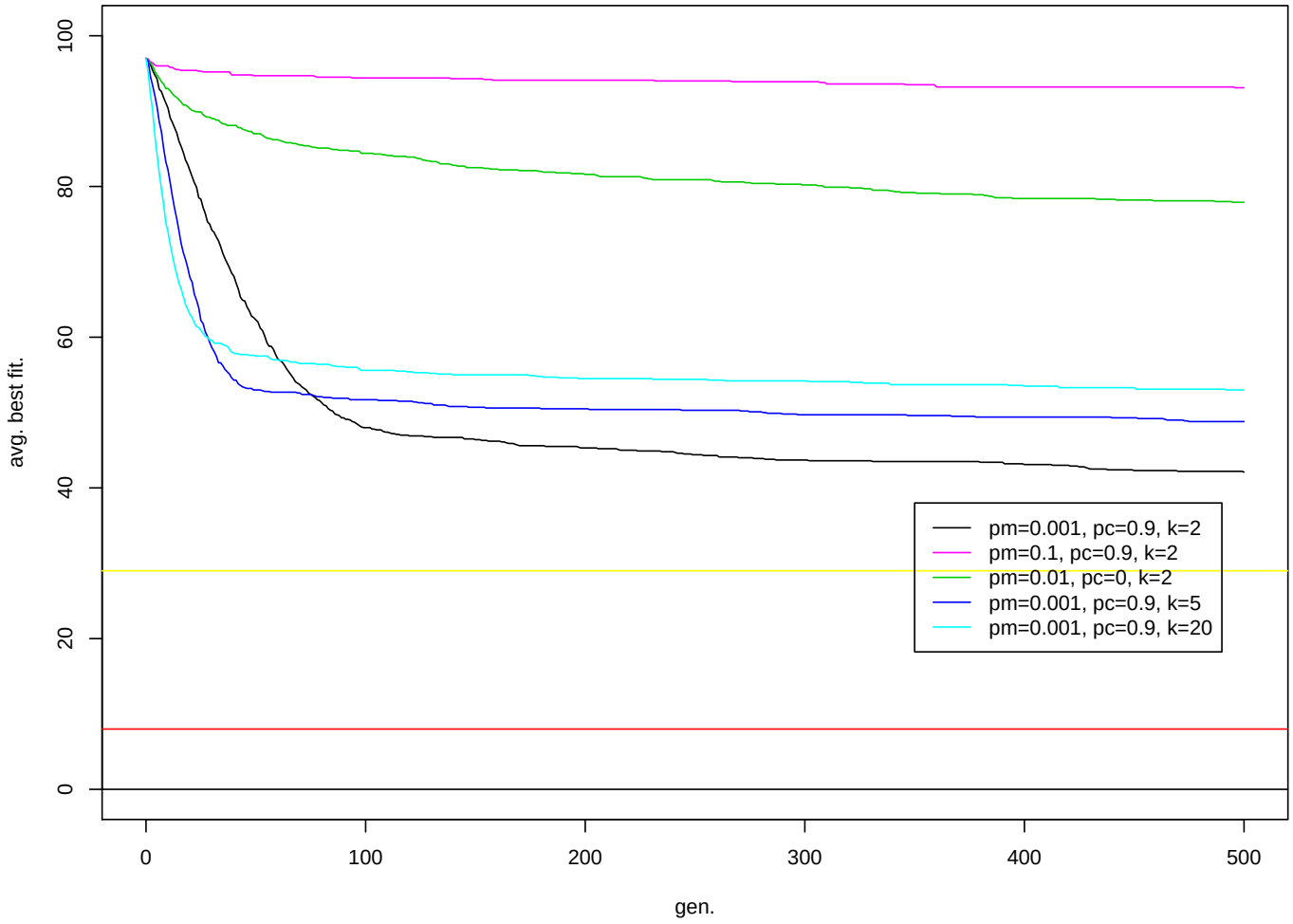
Algoritmo genetico standard:

- Inizializzazione: (pseudo)casuale standard
- Selezione: a torneo di taglia k
- Operatori genetici:
  Crossover con probabilità p_c
  Mutazione con probabilità p_m
- Elitismo (del solo individuo migliore)
- Numero massimo di generazioni G_{max}
- Taglia della popolazione P

Nota: Alcuni parametri (come P, p_m e G_{max}) saranno adattati alla dimensione degli individui.

Algoritmo genetico: prestazioni

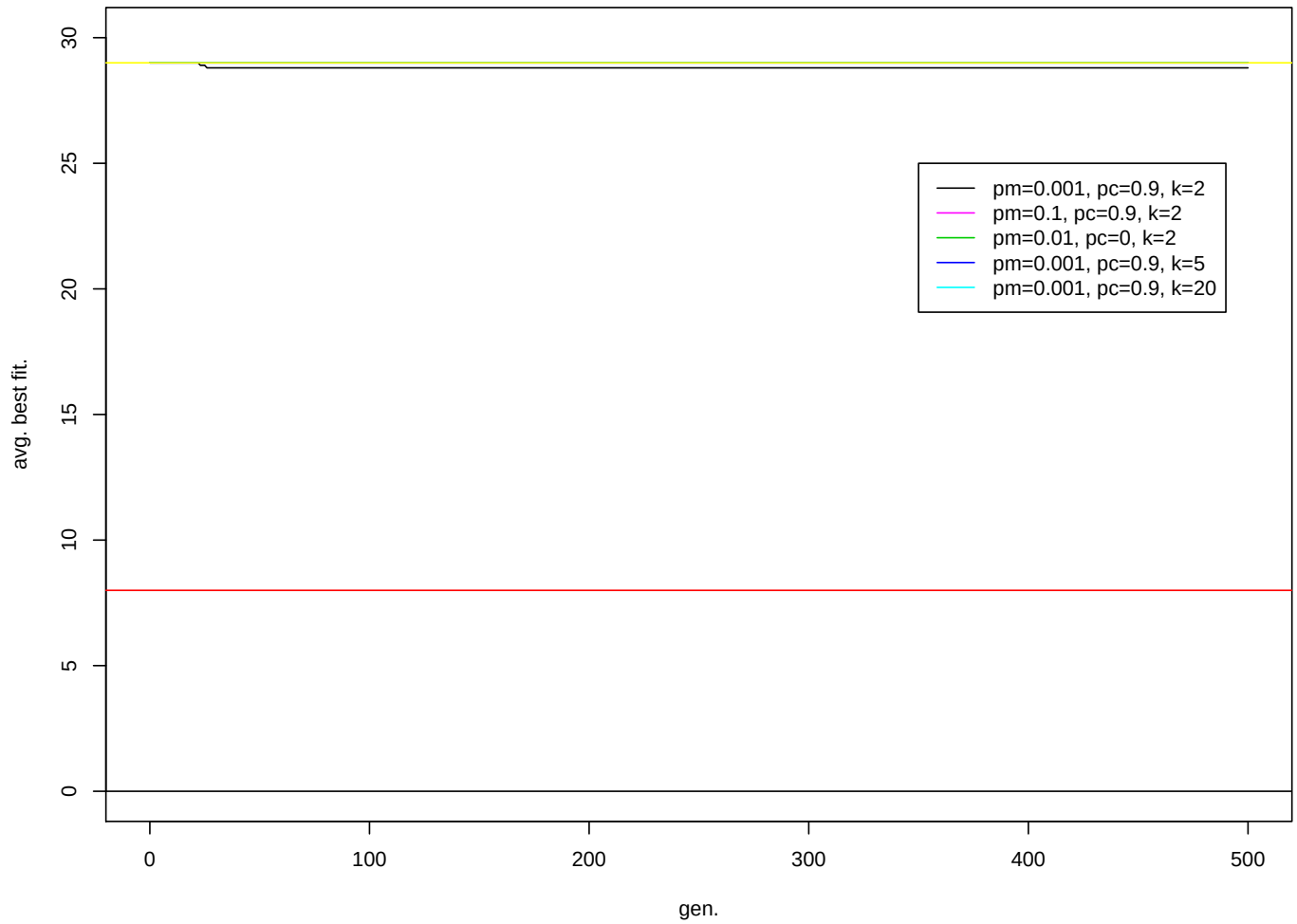
Avg. Best Fitness vs. Generazioni (media di 10 esecuzioni)



$n = 28, \quad m = 8, \quad 2mn = 448, \quad G_{max} = 500, \quad P = 1501$

Algoritmo genetico: prestazioni

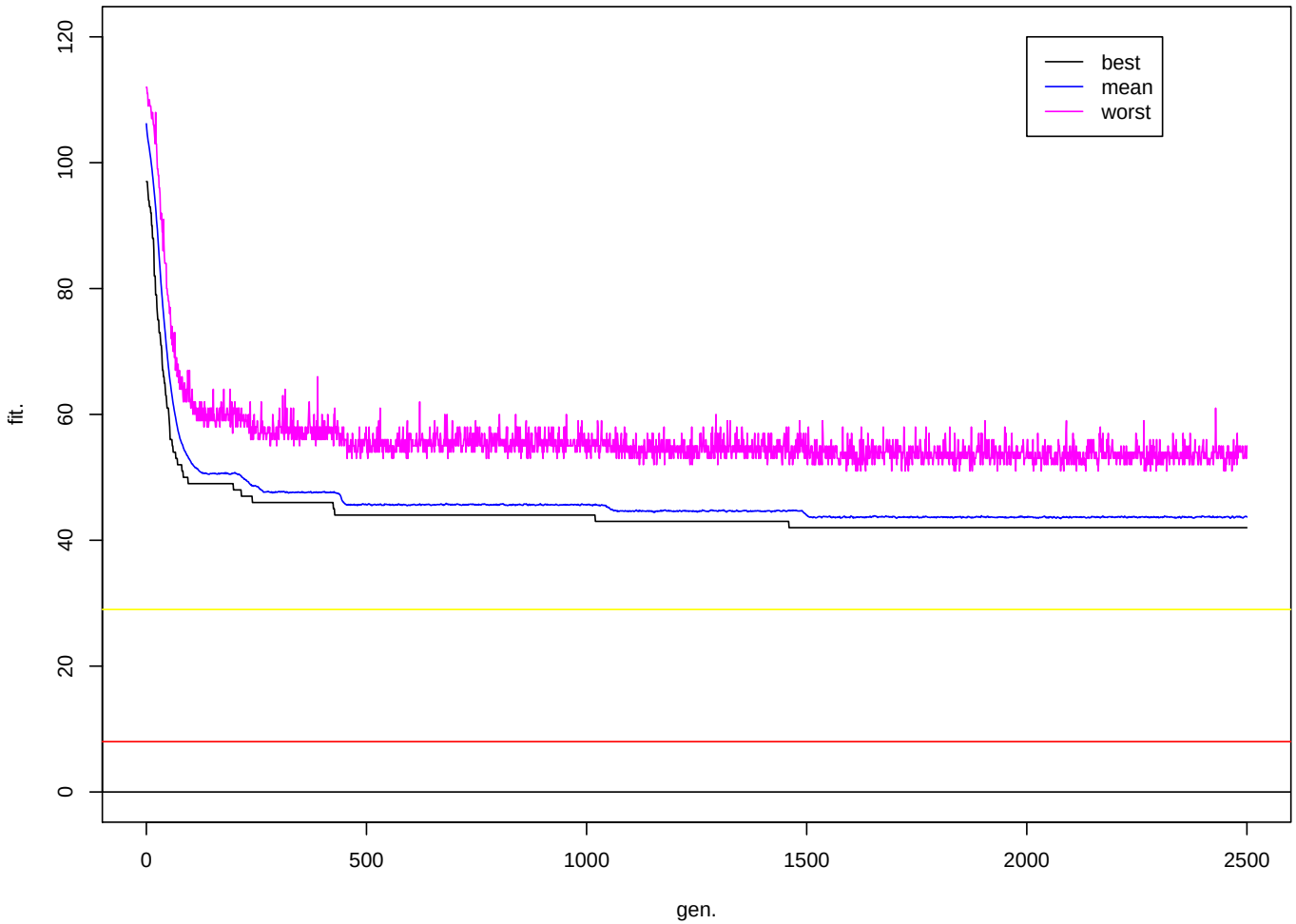
Avg. Best Fitness vs. Generazioni (media di 10 esecuzioni)
con inizializzazione parzialmente guidata



$n = 28, \quad m = 8, \quad 2mn = 448, \quad G_{max} = 500, \quad P = 1501$

Algoritmo genetico: dettaglio di un run

Fitness della popolazione vs. Generazioni (circa 3 min. di esecuzione)



$n = 28, m = 8, 2mn = 448, p_m = 0.001, p_c = 0.9, k = 2, G_{max} = 2500, P = 1501$

Codifica alternativa

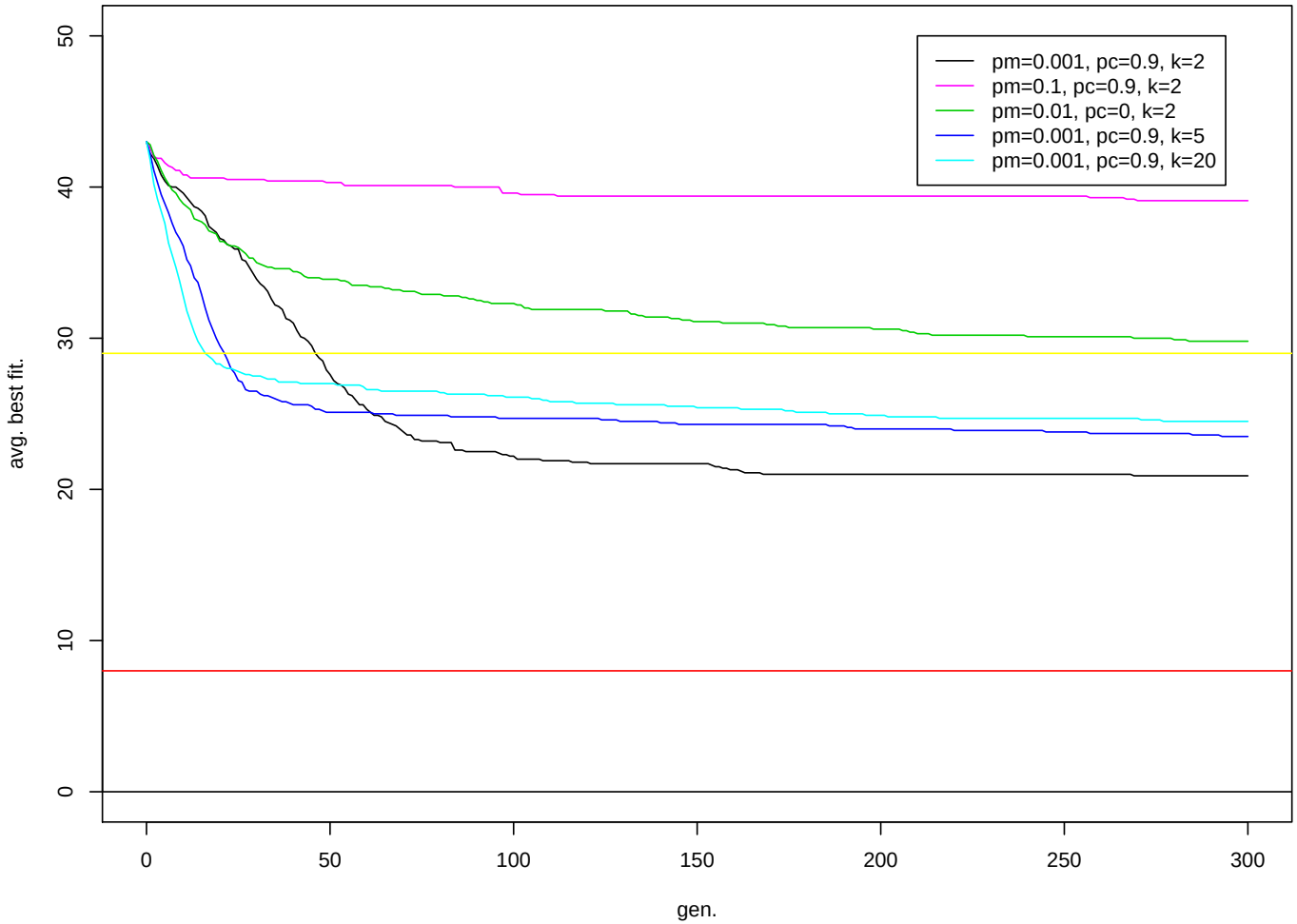
- La codifica "banale" sembra/potrebbe essere svantaggiosa:
  - dipendenza fra posizioni lontane nel vettore
  - individui "inammissibili"
- Codifica senza individui inammissibili:
  - matrice binaria S di dim. $n \times m$ che "dice" come risolvere ciascun sito.

$X[i,j]$	$S[i,j]$	$H[2i,j]$	$H[2i+1,j]$
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

L'individuo è più corto (nm) e non vi sono penalty.

Codifica alternativa: prestazioni

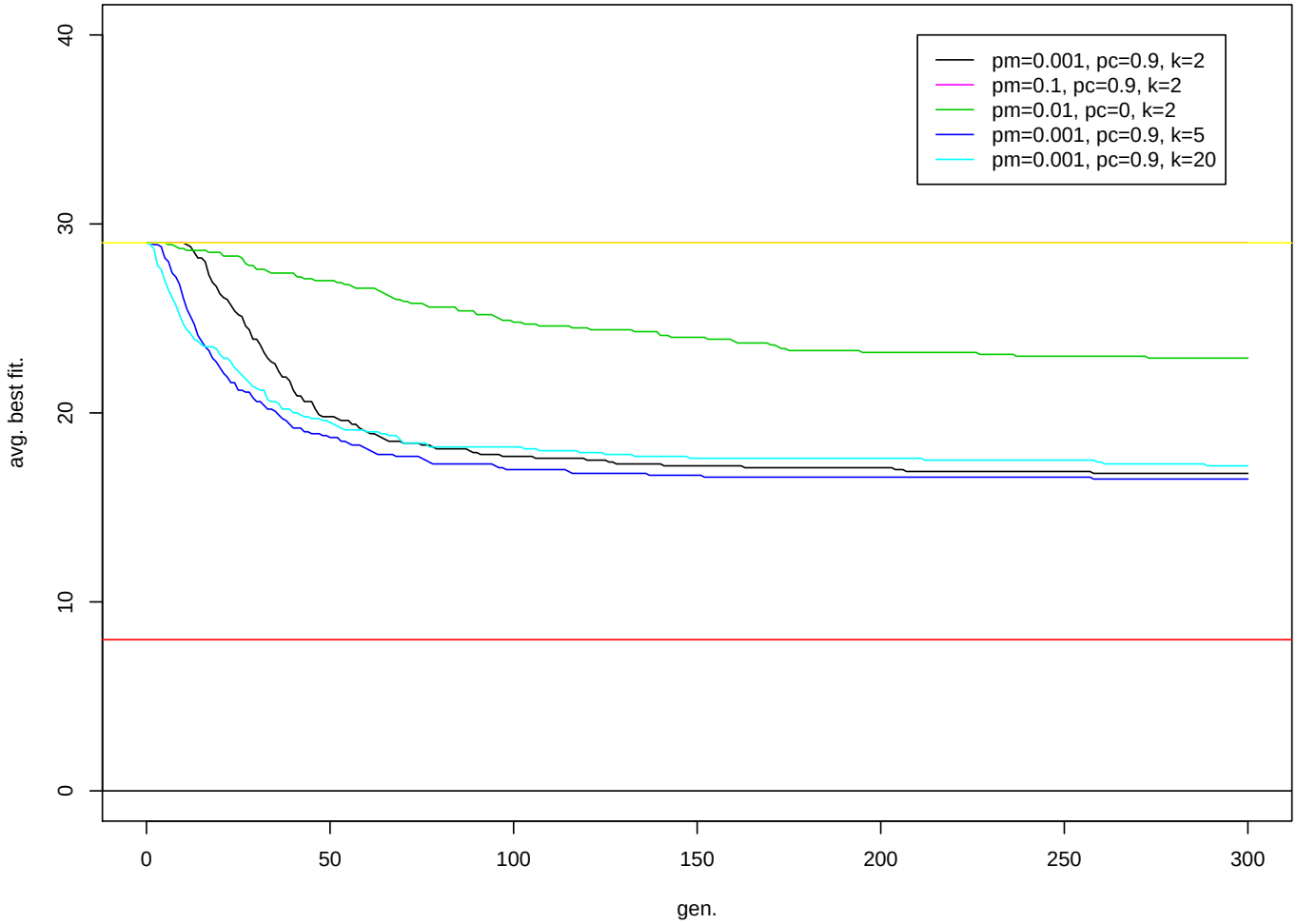
Avg. Best Fitness vs. Generazioni (media di 10 esecuzioni)



$n = 28, \quad m = 8, \quad mn = 224, \quad G_{max} = 300, \quad P = 1501$

Codifica alternativa: prestazioni

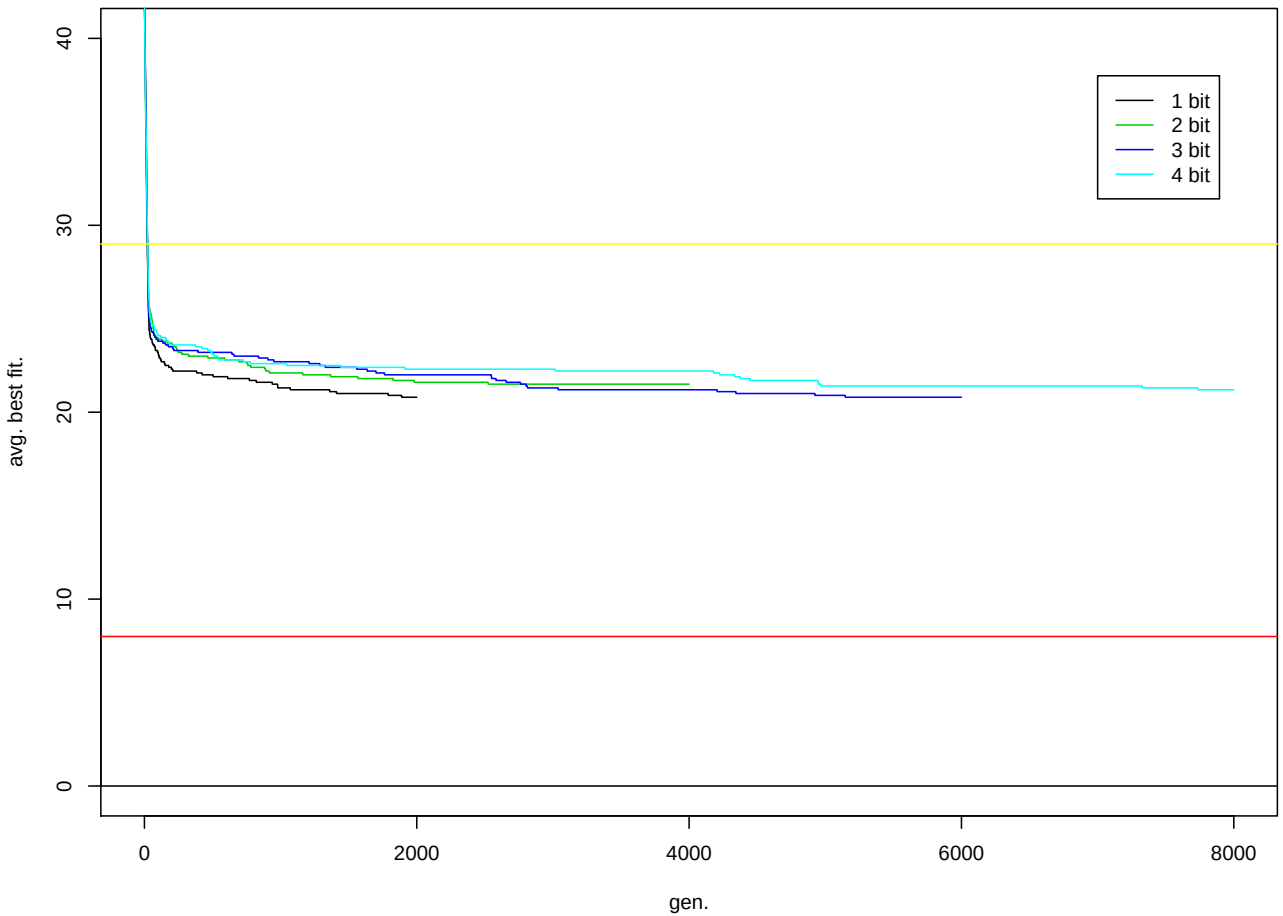
Avg. Best Fitness vs. Generazioni (media di 10 esecuzioni)
con inizializzazione parzialmente guidata



$n = 28, \quad m = 8, \quad mn = 224, \quad G_{max} = 300, \quad P = 1501$

Codifica alternativa: neutralità

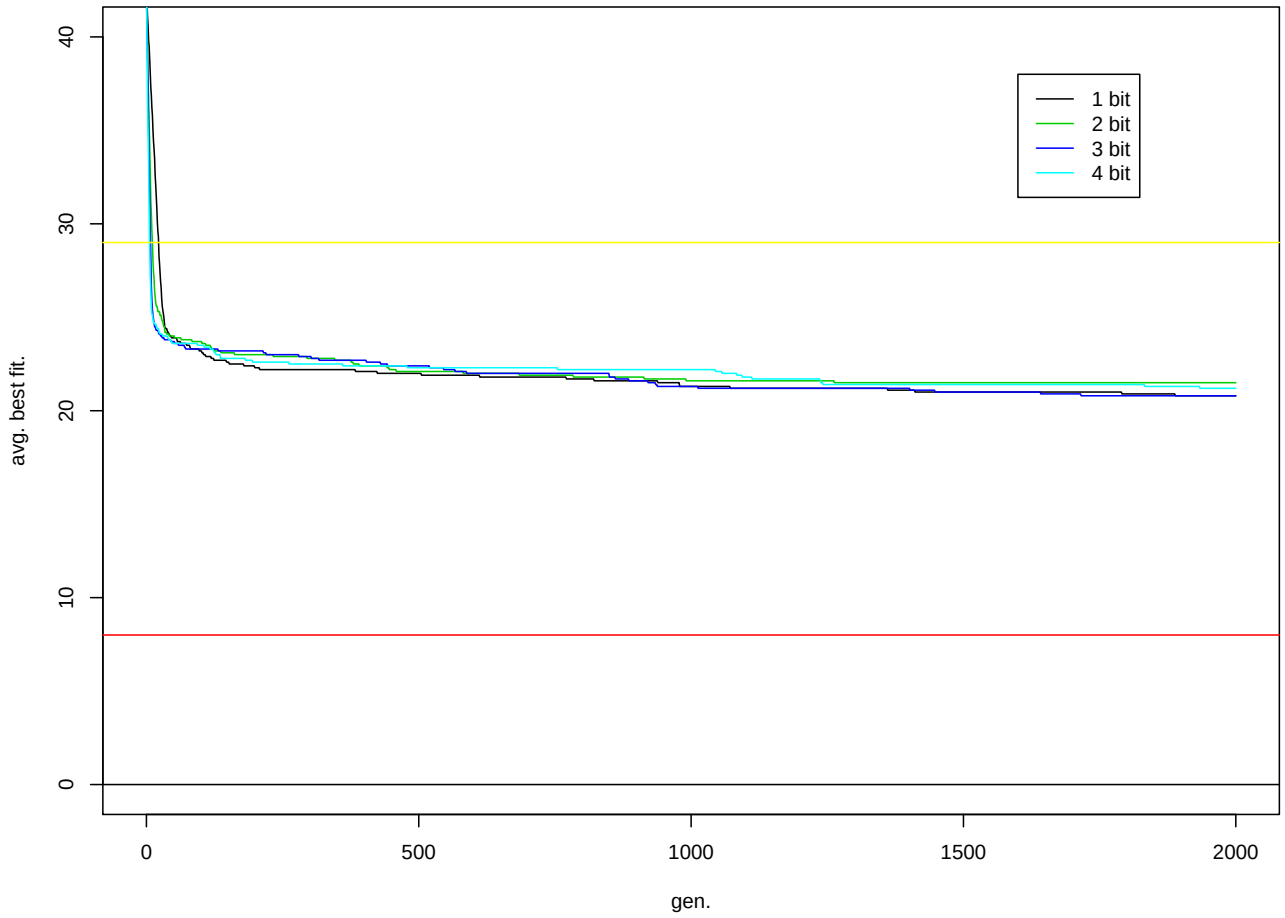
Avg. Best Fitness vs. Generazioni (media di 10 esecuzioni)
con introduzione di "neutralità" artificiale



$n = 28,\ m = 8,\ mn = 224,\ p_m = 0.0005,\ p_c = 0.9,\ k = 4,$
$G_{max} = \text{n. bit} \cdot 2000,\ P = 1501$

Codifica alternativa: neutralità

Avg. Best Fitness vs. Generazioni (compresse) (media di 10 esecuzioni)
con introduzione di "neutralità" artificiale

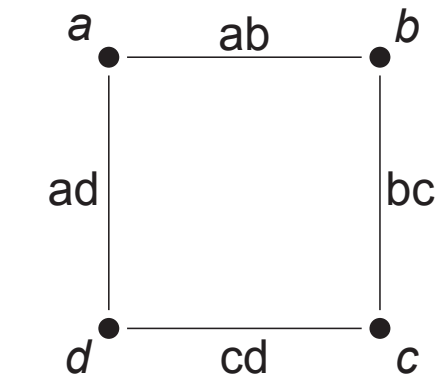


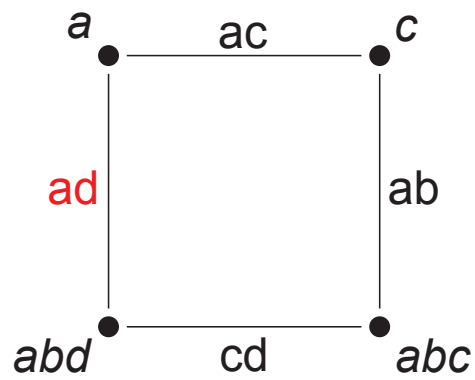
$n = 28,\ m = 8,\ mn = 224,\ p_m = 0.0005,\ p_c = 0.9,\ k = 4,$
$G_{max} = \text{n. bit} \cdot 2000,\ P = 1501$

Xor-grafo

- Lo Xor-grafo è un modello teorico di rappresentazione delle soluzioni.
- *Formalmente:* Sia X un insieme di Xor-genotipi. $X_G = (V, E, \lambda : E \to X)$ è uno Xor-grafo per X sse
  (i) (V, E) è un grafo e
  (ii) λ è iniettiva e
  (iii) per ogni ciclo $\{e_{i_1}, \ldots, e_{i_k}\}$, $\lambda(e_{i_1}) \oplus \ldots \oplus \lambda(e_{i_k}) = \varnothing$.
- Genotipi = Archi, Aplotipi = Vertici
- PPXH: Dato X, trovare uno Xor-grafo X_G per X tale che abbia il minimo numero di vertici.

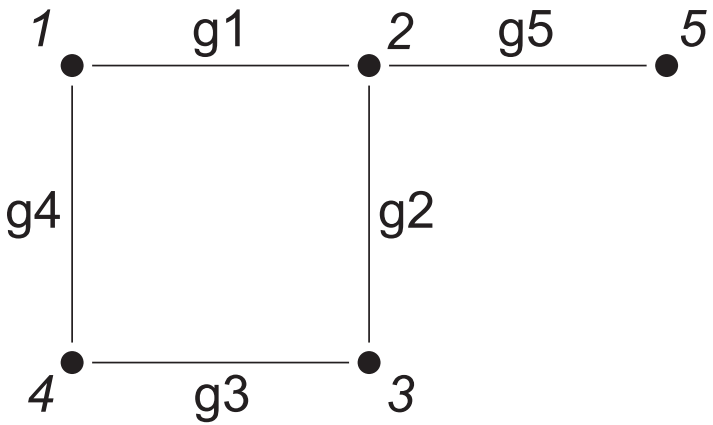
Esempi:





Codifica a Xor-grafo

- Idea: dato X usare AG per trovare lo Xor-grafo di minima cardinalità.
- Individuo = Xor-grafo = sequenza di n elementi di $[0, n]^2$.
- Esempio:

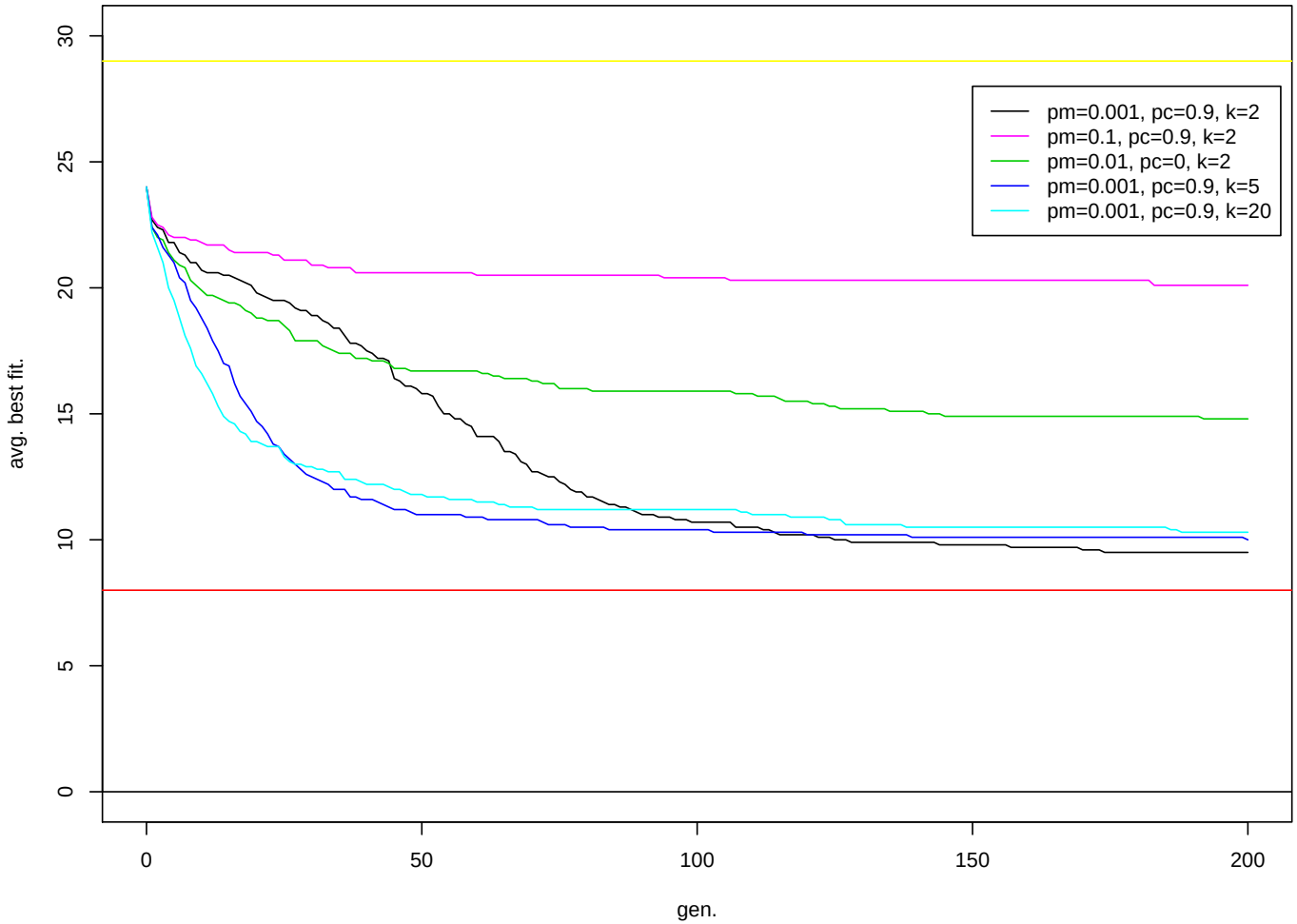


genotipo	v_i	v_j
g_1	1	2
g_2	2	3
g_3	3	4
g_4	4	1
g_5	2	5

- Individuo (sintassi)= sequenza di $2 \cdot \lceil \log_2(n+1) \rceil \cdot n$ bit
- *Non tutti gli individui sono ammissibili!* $\longrightarrow$ Penalty

Codifica a Xor-grafo: prestazioni

Avg. Best Fitness vs. Generazioni (media di 10 esecuzioni)



$n = 28, \quad m = 8, \quad \text{size} = 280, \quad G_{max} = 200, \quad P = 1501$

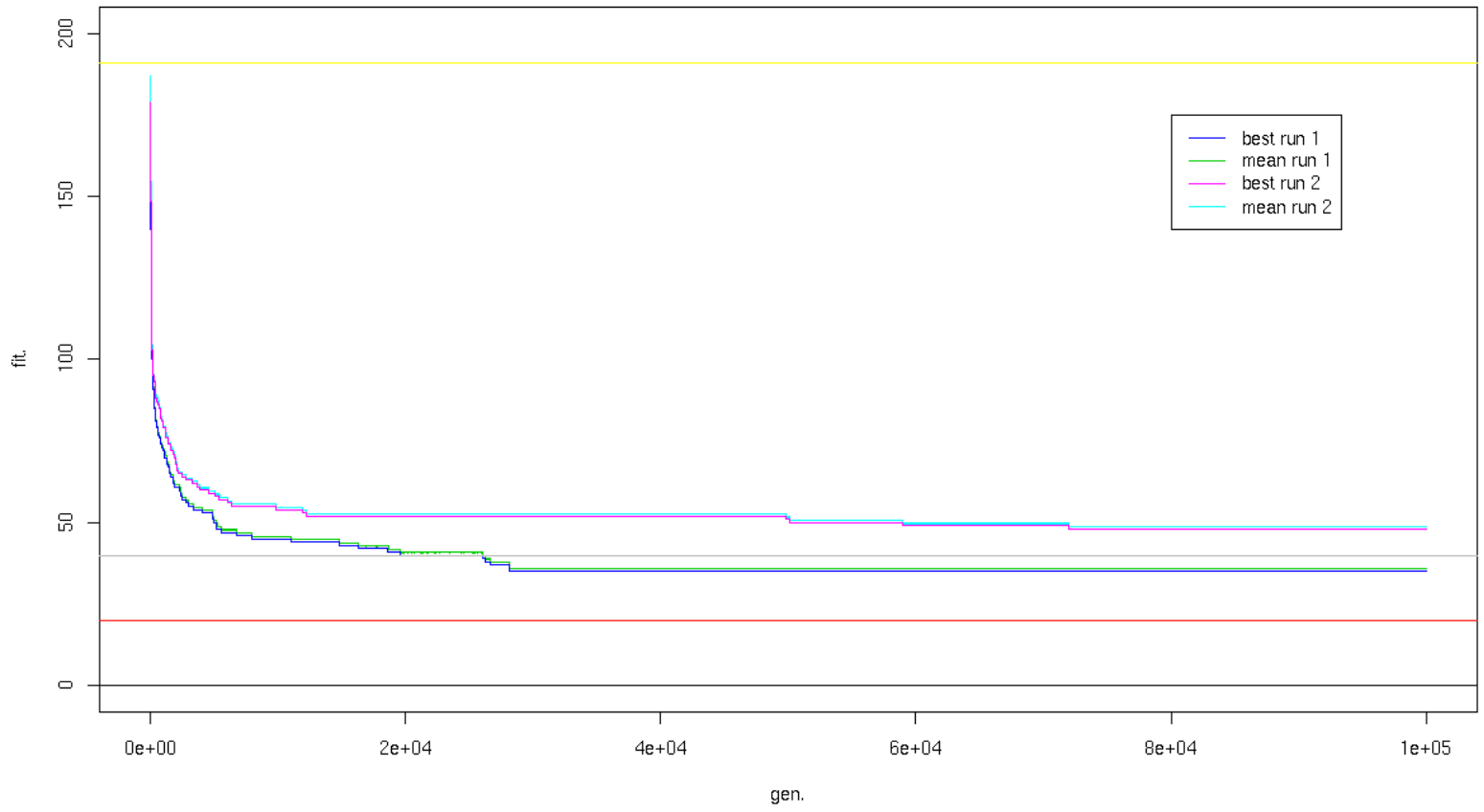
Codifica a Xor-grafo: considerazioni

p_m	p_c	k	success rate
0.001	0.9	2	0.6
0.1	0.9	2	0.0
0.01	0	2	0.0
0.001	0.9	5	0.6
0.001	0.9	20	0.1

- Tempo medio di esecuzione di un run = 13s.
- $|\mathcal{S}(X)| = ((n+1) \cdot n)^n$ (circa).
- Raggiunge l'ottimo e usa meno risorse computazionali.
- Scalabilità del risultato?

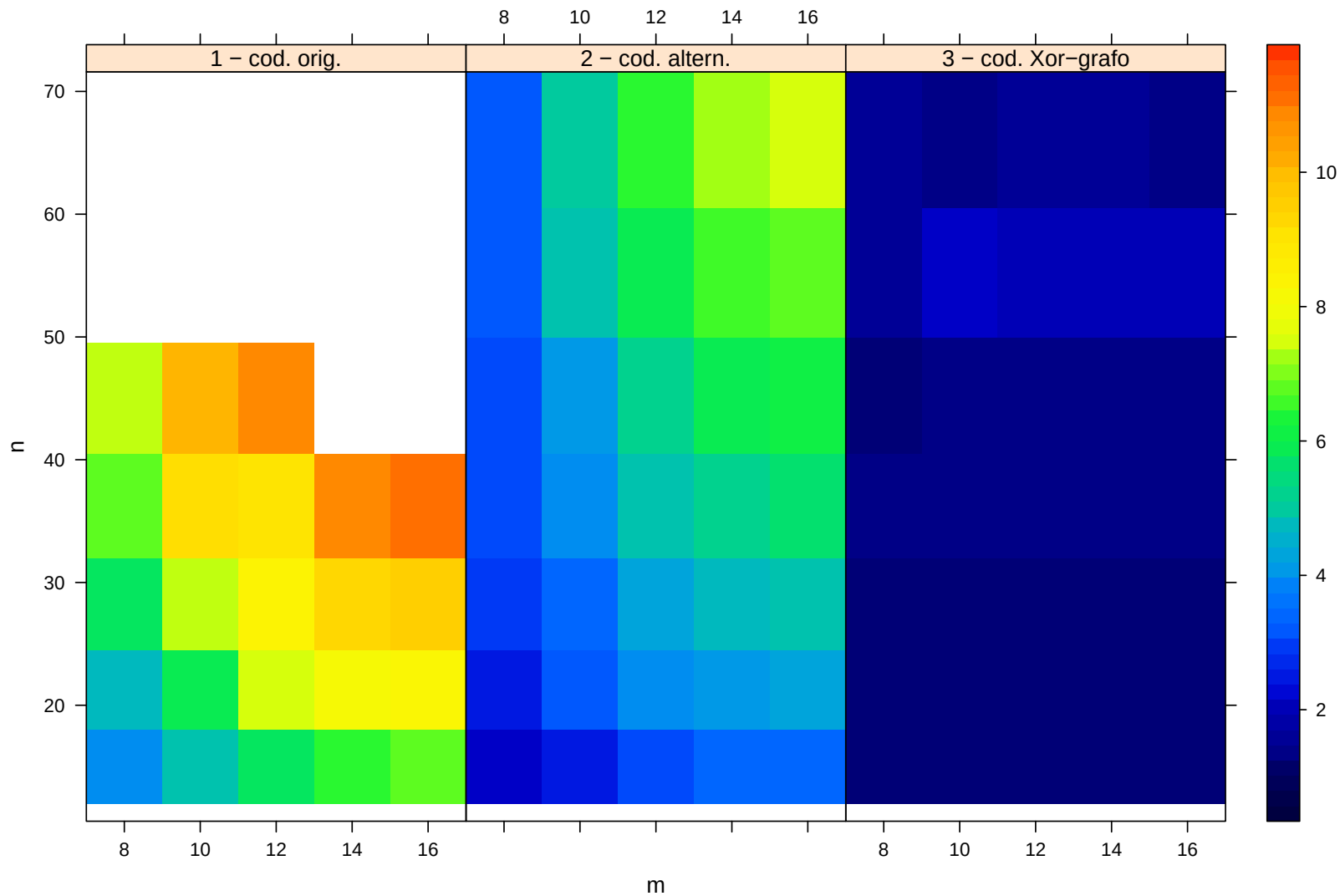
Codifica a Xor-grafo: "grandi" istanze

Fitness della popolazione vs. Generazioni (circa 157 ore di esecuzione)



$n = 190, m = 20, size = 3040, p_m = 0.0001, p_c = 0.9, k = 4, G_{max} = 10^5, P = 10001$

Scalabilità dei risultati



Parametri variabili in funzione della lunghezza degli individui

Conclusioni

- La codifica influenza fortemente risultati e prestazioni!
  - Cod. alternativa riduce gli effetti di dipendenza ed elimina la penalty.
  - Cod. a Xor-grafo introduce "conoscenza del dominio".
  - Cod. a Xor-grafo indipendente da m.
- Settaggio standard dei parametri è il miglior compromesso.
- Inizializzazione guidata non sempre aiuta.
- Convergenza a ottimi locali.
- Gli AG sono utili per PPXH?
  - Richiedono ingenti sforzi computazionali.
  - Nessun algoritmo efficiente (esatto o d'approssimazione) noto.
  - $\Rightarrow$ Non c'è altra scelta! (quasi)